IN THE CLAIMS:

A listing of the status of all claims 1-20 in the present patent application is provided below.

1 (Previously Presented). A parallelizable integrity-aware encryption method, the method comprising the steps of:

whitening at least one message block with a first mask value;

encrypting the at least one whitened message block using a block cipher and a first key; and

whitening the at least one encrypted message block with a second mask value, which is not identical to the first mask value, to generate at least one corresponding output ciphertext block;

wherein the first mask value is computed by applying a XOR function to a first value derived from a NONCE value and a second value derived from encrypting a third value using the block cipher and a second key, and then applying a substitution function to the result of the XOR function;

wherein the first and second key have different values;

wherein the second mask value is computed by applying a XOR function to a fourth value derived from the NONCE value and a fifth value derived from encrypting a sixth value using the

block cipher and the second key, and then applying the substitution function to the result of the XOR function.

2 (Cancelled)

3 (Previously Presented). The method of claim 1, wherein the first and fourth values derived from the NONCE value are permutations of a binary value computed by encrypting the NONCE value using the block cipher and the first key.

4 (Previously Presented). The method of claim 1, wherein the third and sixth values are unique counter values or random numbers.

5 (Previously Presented). The method of claim 1, wherein the steps of whitening each comprise the step of applying a XOR function.

6 (Original). The method of claim 1, further comprising the steps of:

applying a XOR function to all message blocks of a message to compute a XOR-sum;

applying a third mask value to the XOR-sum;

encrypting the masked XOR-sum using the block cipher and the first key; and

applying a fourth mask value to the encrypted XOR-sum to generate an integrity tag.

7 (Previously Presented). The method of claim 6, wherein the third mask value is computed by applying a XOR function to a first value derived from a NONCE value and a second value derived from encrypting a third value using the block cipher and a second key, and then applying a substitution function to the result of the XOR function, wherein the fourth mask value is computed by applying a XOR function to a fourth value derived from the NONCE value and a fifth value derived from encrypting a sixth value using the block cipher and the second key, and then applying the substitution function to the result of the XOR function.

8 (Previously Presented). The method of claim 1, further comprising the steps of:

whitening the at least one output ciphertext block with the second mask value;

decrypting the at least one whitened ciphertext block using a block cipher and the first key; and

whitening the at least one decrypted block with the first mask value to generate at least one corresponding message block.

9 (Original). The method of claim 1, wherein the block cipher is selected from the group consisting of: an Advanced Encryption Standard (AES) block cipher, a Data Encryption Standard (DES) block cipher, and a Triple Data Encryption Standard (3DES) block cipher.

10 (Previously Presented). The method of claim 1, wherein the second and fifth values are elements of a vector.

11 (Previously Presented). At least one processor readable medium for storing a computer program of instructions configured to be readable by at least one processor for instructing the at least one processor to execute a computer process for performing the method as recited in claim 1.

12 (Original). A parallelizable integrity-aware encryption method, the method comprising the steps of:

applying a XOR function to all blocks of a message to compute a XOR-sum;

applying a first mask value to the XOR-sum;

encrypting the masked XOR-sum using a block cipher and a first key; and

applying a second mask value to the encrypted XOR-sum to generate an integrity tag.

13 (Previously Presented). The method of claim 12, wherein the first mask value is computed by applying a XOR function to a first value derived from a NONCE value and a second value derived from encrypting a third value using the block cipher and a second key, and then applying a substitution function to the result of the XOR function, wherein the second mask value is computed by applying a XOR function to a fourth value derived from the NONCE value and a fifth value derived from encrypting a sixth value using the block cipher and the second key, and then applying the substitution function to the result of the XOR function.

14 (Previously Presented). The method of claim 13, wherein the first and fourth values derived from the NONCE value are permutations of a binary value computed by encrypting the NONCE value using the block cipher and the first key.

15 (Previously Presented). The method of claim 12, further

comprising the steps of:

whitening at least one message block with a third mask value;

encrypting the at least one whitened message block using the block cipher and the first key; and

whitening the at least one encrypted message block with the third mask value to generate a corresponding output ciphertext block.

16 (Original). The method of claim 15, wherein the steps of whitening each comprise the step of applying a XOR function.

17 (Original). The method of claim 15, wherein the third mask value is computed by applying a XOR function to a first value derived from a NONCE value and a second value derived from encrypting a third value using the block cipher and a second key, and then applying a substitution function to the result of the XOR function.

18 (Original). The method of claim 12, wherein the block cipher is selected from the group consisting of: an Advanced Encryption Standard (AES) block cipher, a Data Encryption Standard (DES) block cipher, and a Triple Data Encryption Standard (3DES) block

cipher.

19 (Previously Presented). The method of claim 13, wherein the second and fifth values are elements of a vector.

20 (Previously Presented). At least one processor readable medium for storing a computer program of instructions configured to be readable by at least one processor for instructing the at least one processor to execute a computer process for performing the method as recited in claim 12.